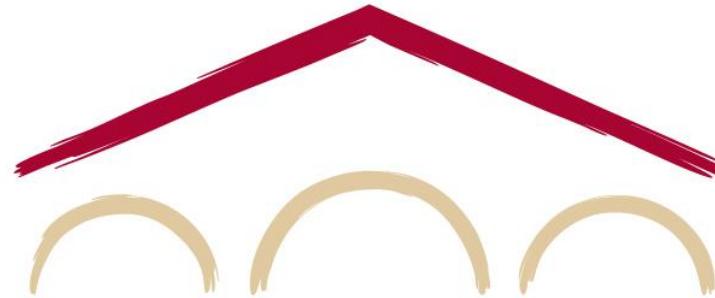


# Natural Language Processing with Deep Learning

**CS224N/Ling284**



Diyi Yang

Lecture 7: Post-training

# Lecture Plan

1. Instruction fine-tuning [15 mins]
2. Reinforcement learning from human preferences (RLHF) [20 mins]
3. InstructGPT and ChatGPT [5 mins]
4. Limitation of RL and reward modeling [5 mins]
5. Introducing Direct Preference Optimization (DPO) [25 mins]
6. Human preference data; human vs. AI Feedback [5 mins]

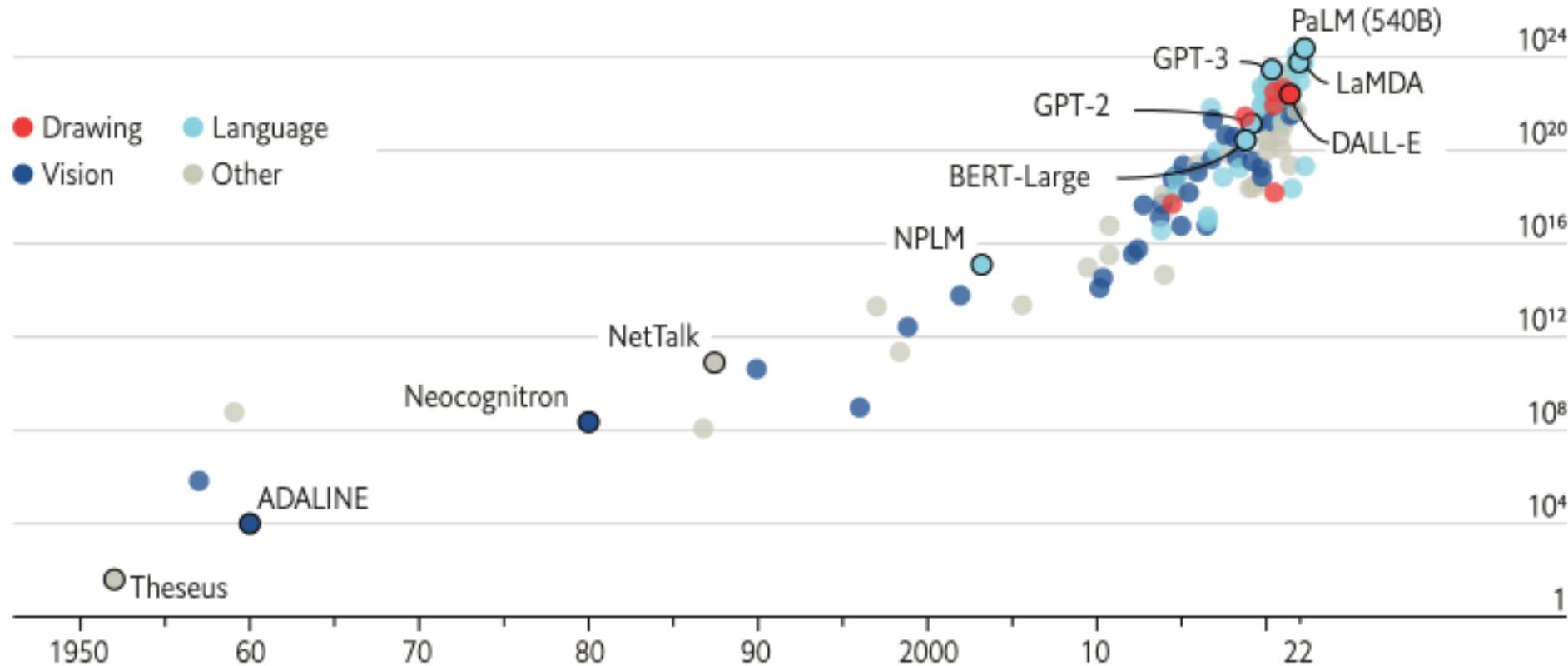
**Project** information comes out today! Find your team ☺ Assignment 3 due on Feb 5<sup>th</sup>.

# Language models getting larger and larger

## The blessings of scale

AI training runs, estimated computing resources used

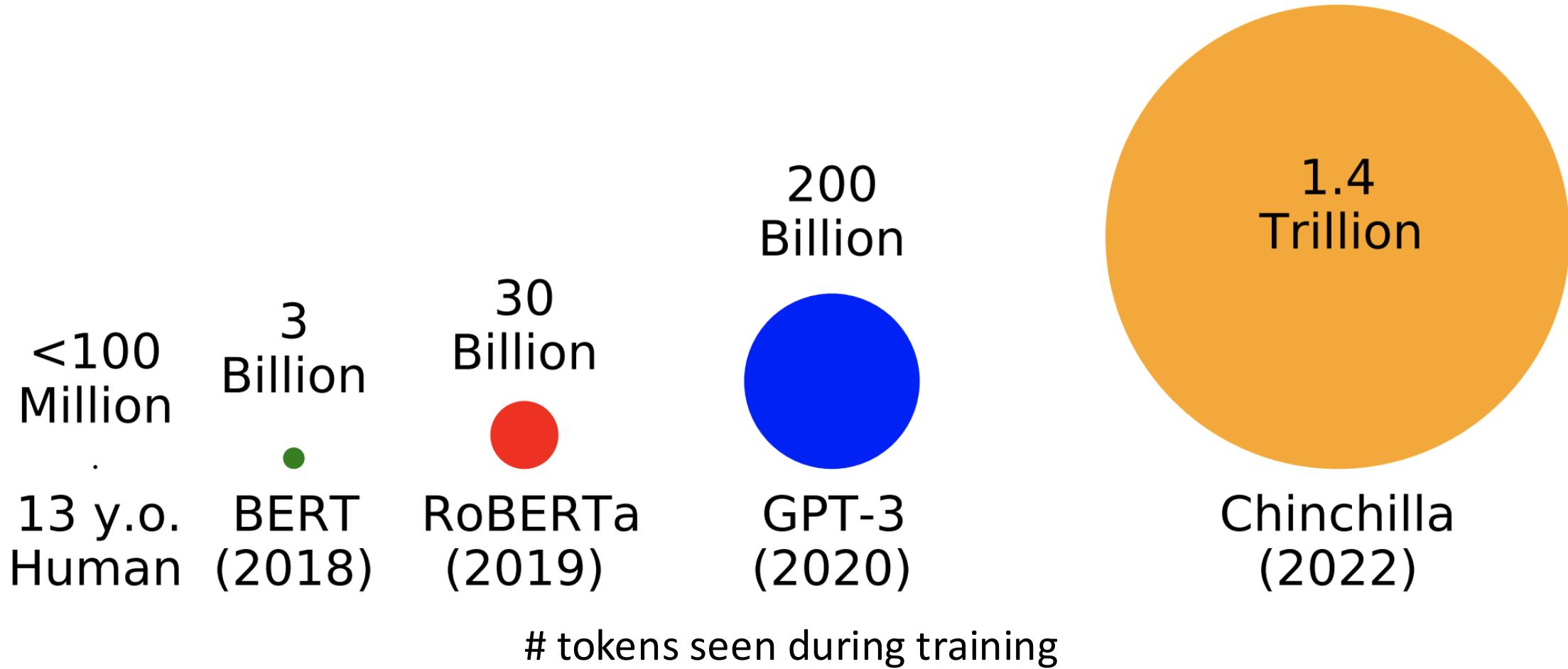
Floating-point operations, selected systems, by type, log scale



Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data

<https://www.economist.com/interactive/briefing/2022/06/11/huge-foundation-models-are-turbo-charging-ai-progress>

# Language models trained on more and more data



<https://babylm.github.io/>

# Language models as world models?

We can describe circles in the  $xy$ -plane using equations in terms of  $x$  and  $y$ . Circle equations questions require us to understand the connection between these equations and the features of circles.

For example, the equation  $(x + 2)^2 + (y - 3)^2 = 4^2$  is graphed in the  $xy$ -plane below. It is a circle with a center at  $(-2, 3)$  and a radius of 4.

<https://www.khanacademy.org/test-prep/sat/x0a8c2e5f:untitled-652>

```
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```

<https://github.com/features/copilot>

**I need to throw a dinner party for 6 people who are vegetarian. Can you suggest a 3-course menu with a chocolate dessert?**

[\[Microsoft Bing\]](#)

(Also see OpenAI's ChatGPT, Anthropic's Claude)

Sure, I can suggest a 3-course menu with a chocolate dessert for your vegetarian dinner party. Here are some possible options:

- Starter: **Wild Mushroom Tartlets with Onion Sauce** <sup>1</sup> or **Vegan Popcorn Tofu Nuggets** <sup>2</sup>
- Main: **Vegan Butternut Squash Mac** <sup>2</sup> or **Vegetarian Three-Bean Chili** <sup>3</sup>
- Dessert: **Chocolate Lava Cake** <sup>4</sup> or **Chocolate Pasta with Chocolate Hazelnut Cream Sauce, White Chocolate Shavings, and Fresh Berries** <sup>4</sup>

I hope this helps you plan your party. Enjoy! 😊

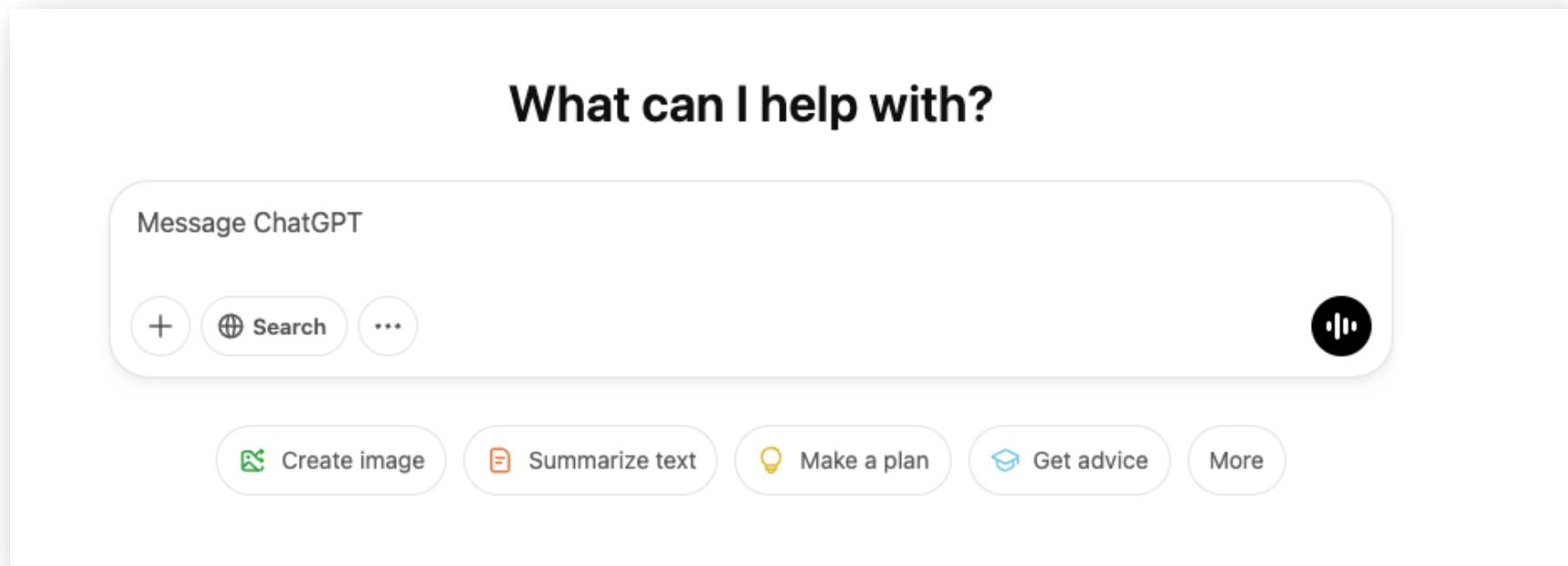
Learn more: [1. booths.co.uk](#) [+10 more](#)

# Language models as multitask assistants?

How do we get from *this*

*Stanford University is located in \_\_\_\_\_*

to *this*?



# Lecture Plan

- 1. Instruction fine-tuning**
2. Reinforcement learning from human preferences (RLHF)
3. InstructGPT and ChatGPT
4. Limitation of RL and reward modeling
5. Introducing Direct Preference Optimization (DPO)
6. Human preference data; human vs. AI Feedback
7. What's next?

# Language modeling ≠ assisting users

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

Language models are not *aligned* with user intent [[Ouyang et al., 2022](#)]

Finetuning to the rescue!

# Language modeling ≠ assisting users

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION **Human**

A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

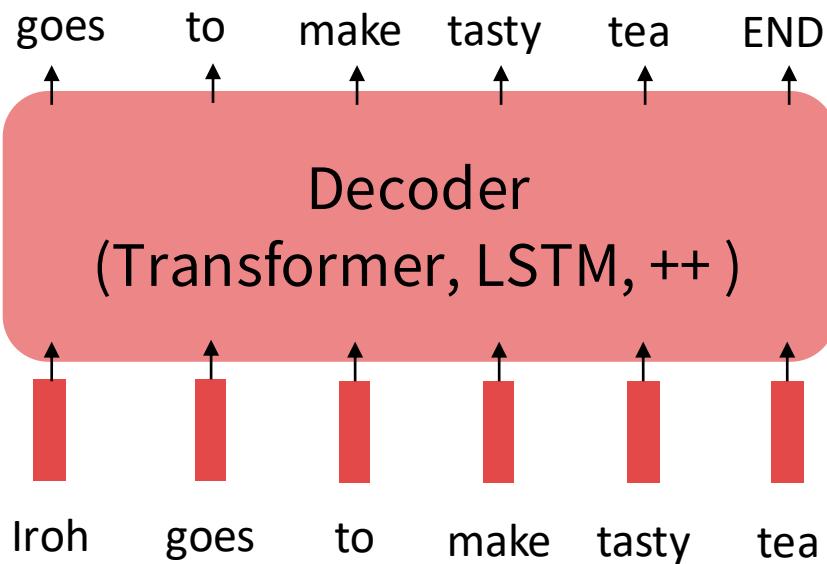
Language models are not *aligned* with user intent [[Ouyang et al., 2022](#)]  
**Finetuning to the rescue!**

# The pretraining/finetuning paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

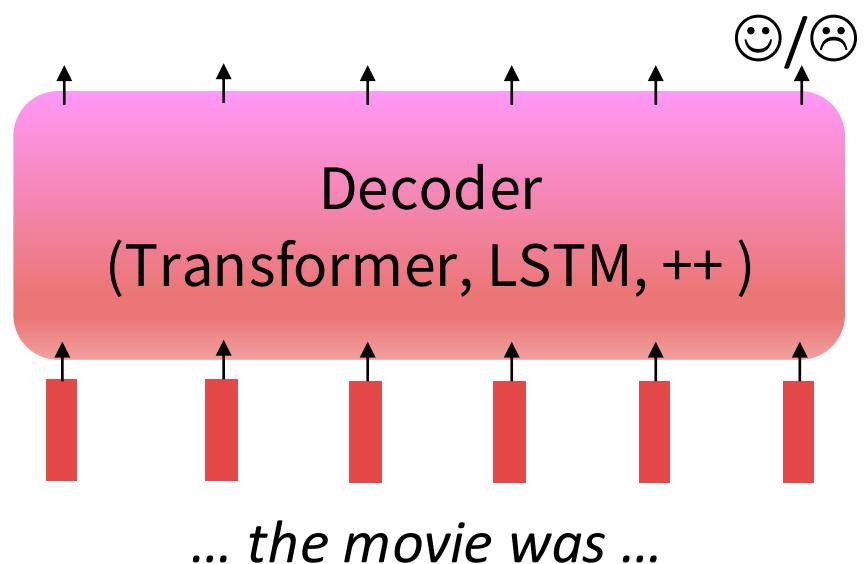
## Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



## Step 2: Finetune (on your task)

Not many labels; adapt to the task!

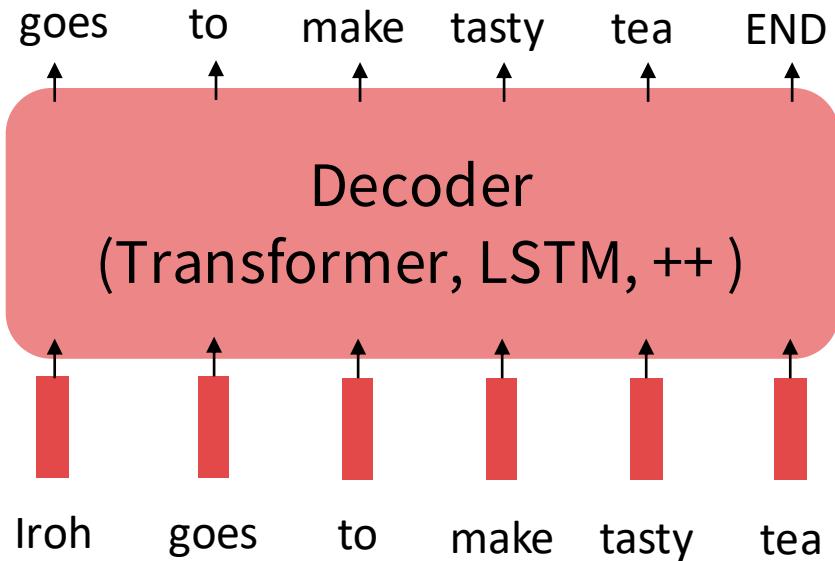


# Scaling up finetuning

Pretraining can improve NLP applications by serving as parameter initialization.

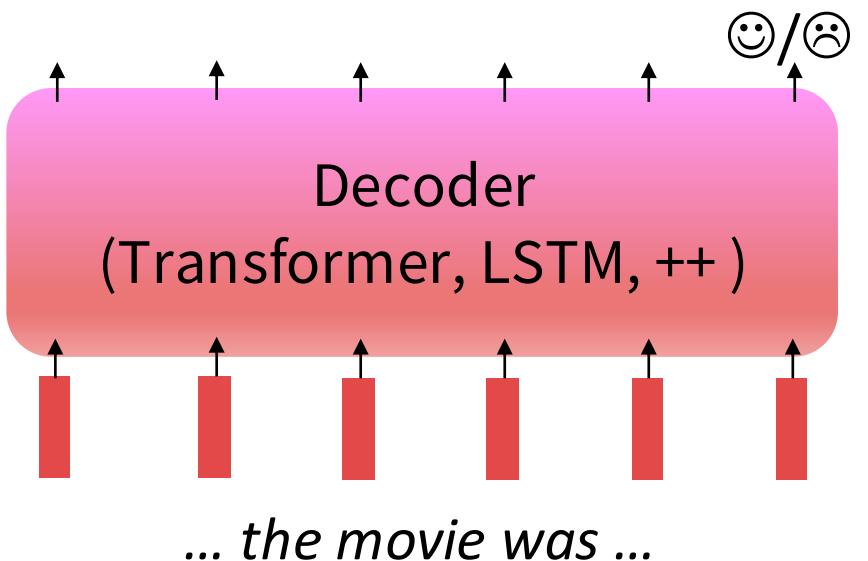
## Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



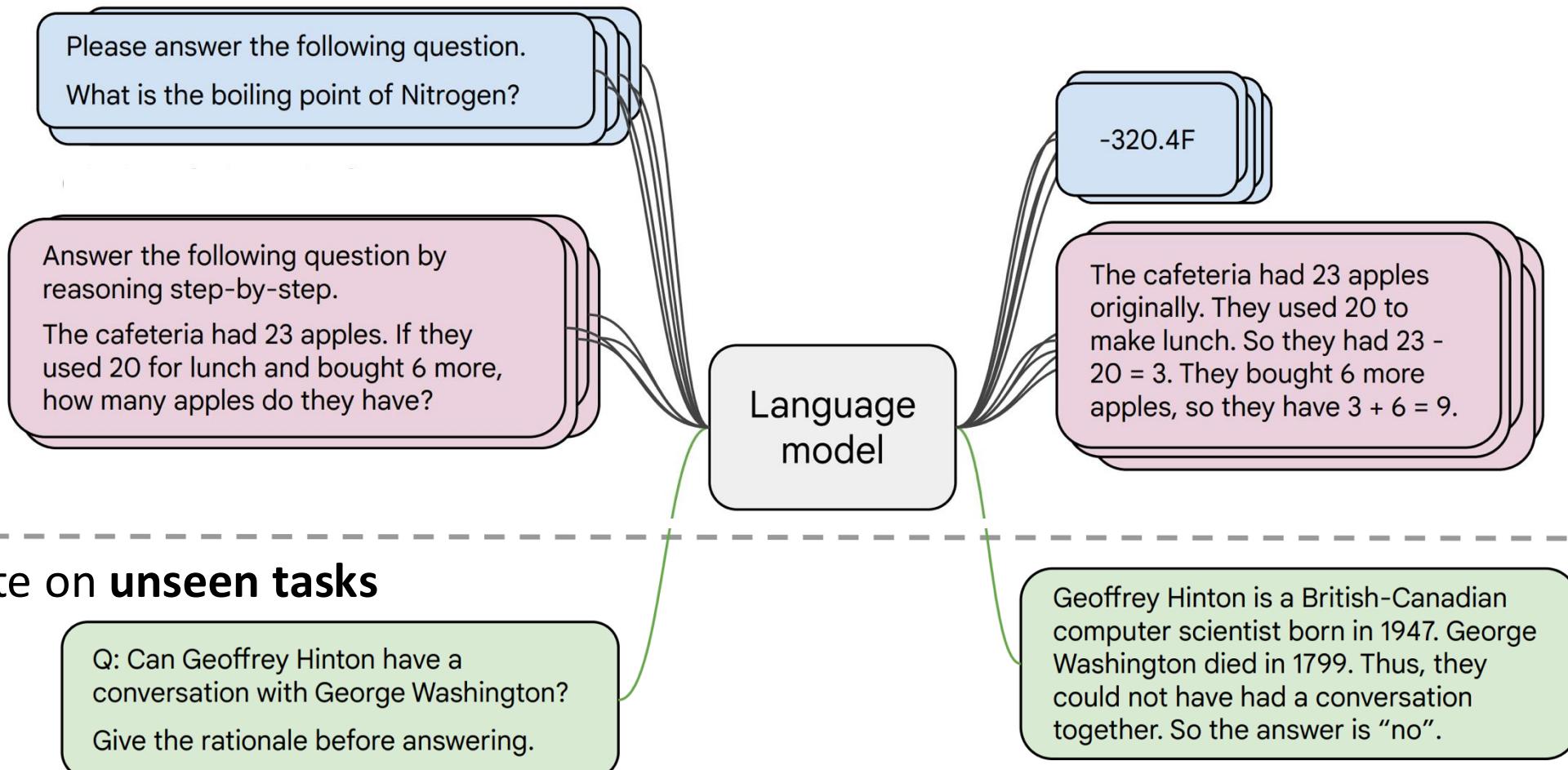
## Step 2: Finetune (on many tasks)

Not many labels; adapt to the tasks!



# Instruction finetuning

- **Collect examples of (instruction, output) pairs across many tasks and finetune an LM**



[FLAN-T5; [Chung et al., 2022](#)]

# Instruction ~~finetuning~~ pretraining?

- As is usually the case, **data + model scale** is key for this to work!
- **Super-NaturalInstructions** dataset contains **over 1.6K tasks, 3M+ examples**
  - Classification, sequence tagging, rewriting, translation, QA...

**Q:** how do we evaluate such a model?

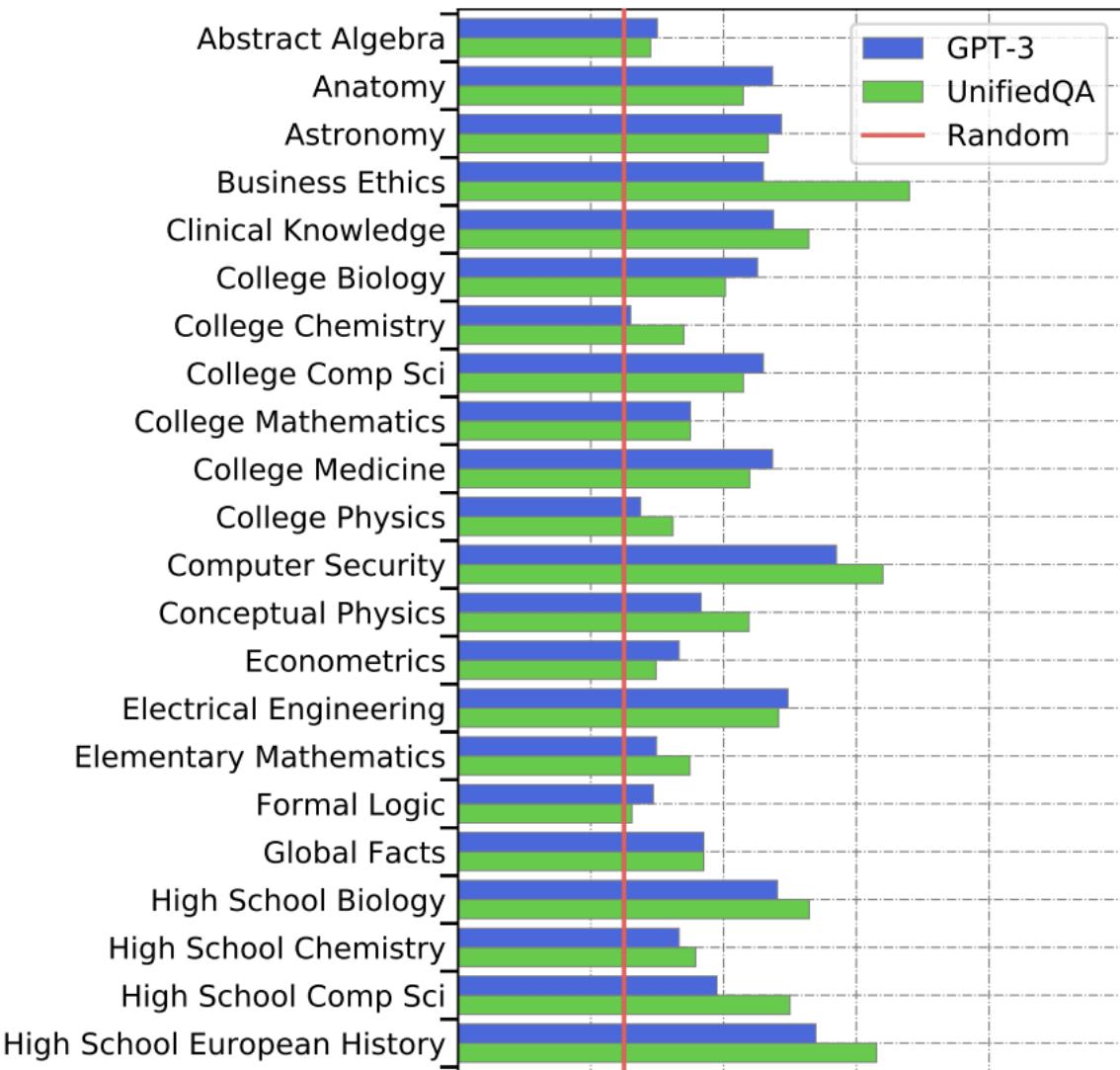


[Wang et al., 2022]

# Aside: new benchmarks for multitask LMs

## Massive Multitask Language Understanding (MMLU) [Hendrycks et al., 2021]

New benchmarks for measuring LM performance on 57 diverse *knowledge intensive* tasks



# Some intuition: examples from MMLU

## Astronomy

What is true for a type-Ia supernova?

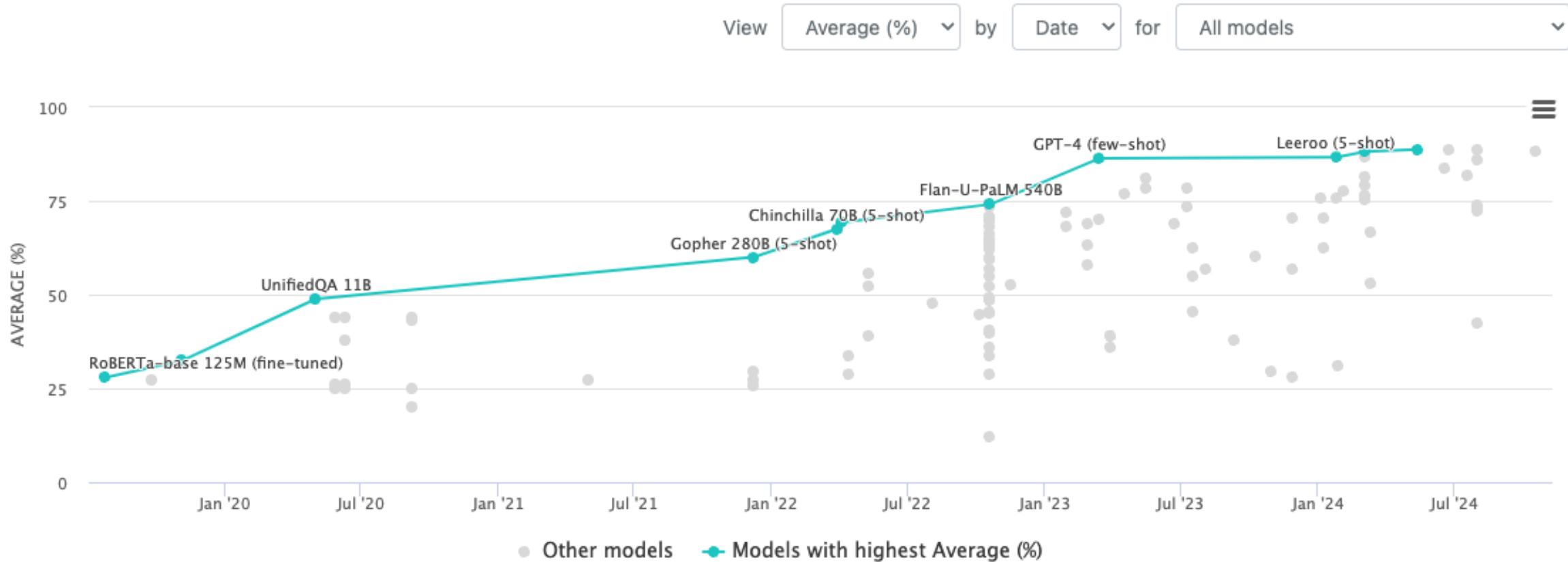
- A. This type occurs in binary systems.
- B. This type occurs in young galaxies.
- C. This type produces gamma-ray bursts.
- D. This type produces high amounts of X-rays.

## High School Biology

In a population of giraffes, an environmental change occurs that favors individuals that are tallest. As a result, more of the taller individuals are able to obtain nutrients and survive to pass along their genetic information. This is an example of

- A. directional selection.
- B. stabilizing selection.
- C. sexual selection.
- D. disruptive selection

# Progress on MMLU



- Rapid, impressive progress on challenging knowledge-intensive benchmarks

## Aside: new benchmarks for multitask LMs

## BIG-Bench [Srivastava et al., 2022]

200+ tasks, spanning:



[https://github.com/google/BIG-bench/blob/main/bigbench/benchmark\\_tasks/README.md](https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/README.md)

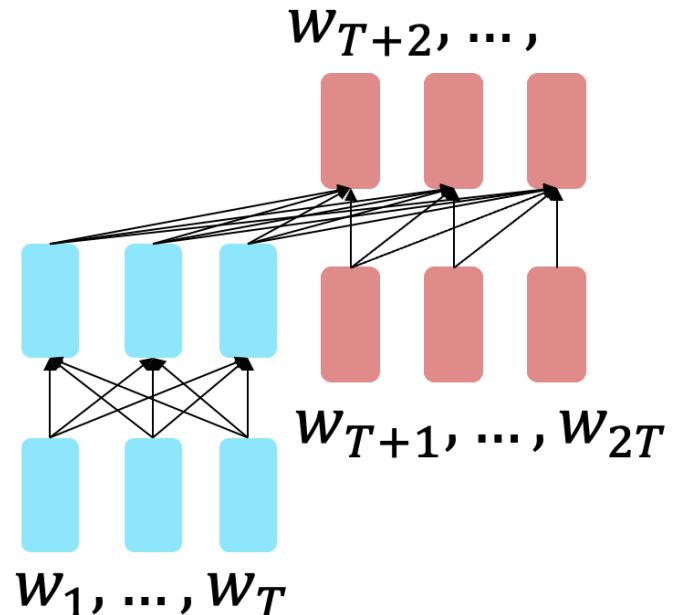
# BEYOND THE IMITATION GAME: QUANTIFYING AND EXTRAPOLATING THE CAPABILITIES OF LANGUAGE MODELS

#### Alphabetic author list:\*

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluza, Aitor Lewkowycz, Aksat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Aliz Tazar, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Aminda Askold, Amanda Snouza, Ambros Sloane, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angelia Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottard, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Arshul Mullokhand, Ashish Sabharwal, Austin Herrick, Ayva Erfat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özburn, Behnam Heydayatian, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Cameron Dia, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkoff, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sile, Dan Garrett, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguf González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debayoti Datta, Deep Guha, Dennis Zeng, Denis Klekyo, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupsels, Digantha Misra, Dilyar Buzan, Dimitri Coelho Mora, Diyi Yang, Dong-Ho Lee, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ilie Pavlicek, Hanuèle Rodola, Emma Lam, Eric Chu, Eric Tang, Ertuğrul Erdem, Enric Chang, Ethan A. Chi, Ethan Dyer, Ethan Jheraz, Ethan Kim, Eunice Enegefu Manyasi, Evigeni Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovich-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannam Kim, Hannah Rashkin, Hamaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevin, Hinrich Schütze, Hiromi Yakuwa, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kermion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppell, James Zheng, James Zou, Jan Kocon, Jana Thompson, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Boschker, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xiong, Jianming Song, Jillian Tang, Joan Wawer, John Burden, John Miller, John U. Balis, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Budenbaum, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kancerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kausubh D. Dhole, Kevin Gimpel, Kevin Omundi, Kory Mathewson, Kristen Chiaffuoli, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfü Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Maanaa Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Mara Jose Ramírez Quintana, Marie Tolkehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátýás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitsky, Michael Starrit, Michael Strubel, Michael Swedrowski, Michel Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mihuee Xu, Mirac Suzgun, Mo Tiwari, Mohit Bansal, Moin Amminaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nikita Nangia, Niklas Deckers, Niklas Muenninghoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuwan Weni, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Morena Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Wang, Piotr Milkowski, Piyush Patil, Pouya Pezeshkhani, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Robin Banjade, Rachel Etta Rudolph, Rafeer Gabriel, Rafael Habacker, Ramón Risco Delgado, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbie Raymackers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitlev, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Tenehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajad Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Grueter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shang Gu, Shubhi Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolina (Shammie) Debnath, Sihamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sreethara Hatwar, Stanislas Dehaene, Steven Dvivc, Stefano Ermont, Stéphane Biderman, Stephanie Lin, Stephen Prasad, Steven T. Plantadosi, Stuart M. Shieber, Summer Mischerghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Taro Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Thé Desbordes, Theodore Rothchild, Thomas Phan, Tianle Wang, Tiberius Nkonyili, Timo Schick, Timofei Kornev, Timothy Telleken-Lawton, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wong Yossen, Xiang Ren, Xiaoyu Tong, Xinxian Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghobzadeh, Yair Lazkretz, Yangqiu Song, Yasaman Bai, Yejin Choi, Yichang Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yusang Hou, Yuntao Bai, Zachary Seid, Zhuoyue Zhao, Zijian Zhi, Zirui Wang, Zivui Wu

# Instruction finetuning and performance gains

- Recall the T5 encoder-decoder model [[Raffel et al., 2018](#)], pretrained on the **span corruption** task
- Flan-T5** [[Chung et al., 2022](#)]: T5 models finetuned on 1.8K additional tasks



Params	Model	<b>BIG-bench + MMLU</b> Norm. avg.
80M	T5-Small	-9.2
	Flan-T5-Small	-3.1 <b>(+6.1)</b>
250M	T5-Base	-5.1
	Flan-T5-Base	6.5 <b>(+11.6)</b>
780M	T5-Large	-5.0
	Flan-T5-Large	13.8 <b>(+18.8)</b>
3B	T5-XL	-4.1
	Flan-T5-XL	19.1 <b>(+23.2)</b>
11B	T5-XXL	-2.9
	Flan-T5-XXL	23.7 <b>(+26.6)</b>

**Bigger model = bigger  $\Delta$**

# Instruction finetuning and performance gains

## Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:

- (A) They will discuss the reporter's favorite dishes
- (B) They will discuss the chef's favorite dishes
- (C) Ambiguous

A: Let's think step by step.

## Before instruction finetuning

The reporter and the chef will discuss their favorite dishes.

The reporter and the chef will discuss the reporter's favorite dishes.

The reporter and the chef will discuss the chef's favorite dishes.

The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

✖ (doesn't answer question)

# Instruction finetuning and performance gains

## Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:

- (A) They will discuss the reporter's favorite dishes
- (B) They will discuss the chef's favorite dishes
- (C) Ambiguous

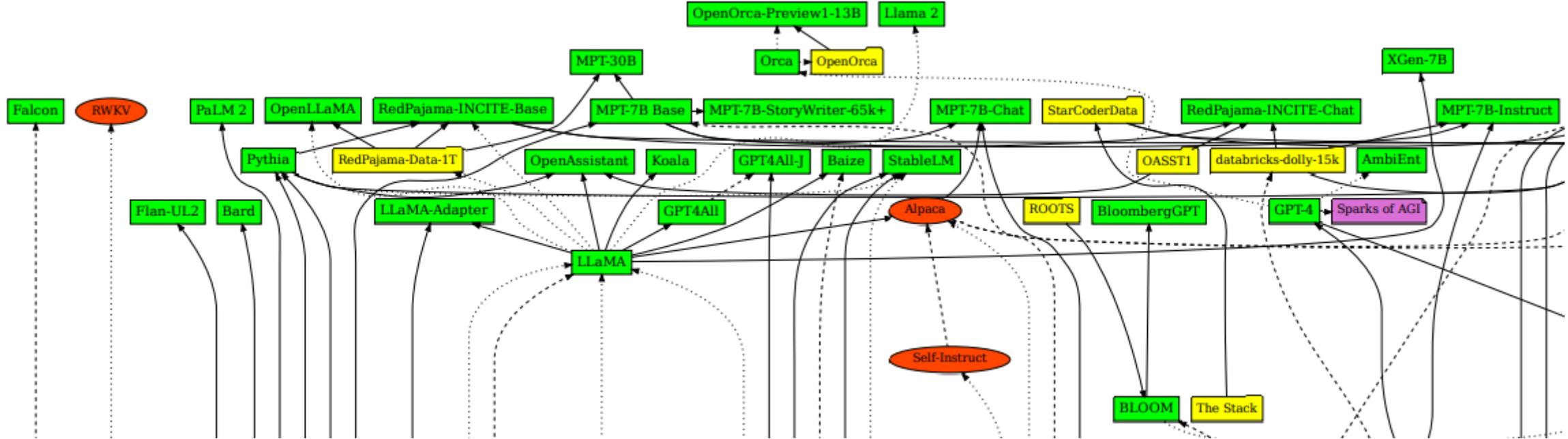
A: Let's think step by step.

## After instruction finetuning

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). 

Try FLAN-T5 out to get a sense of its capabilities: <https://huggingface.co/google/flan-t5-xxl> [Chung et al., 2022]

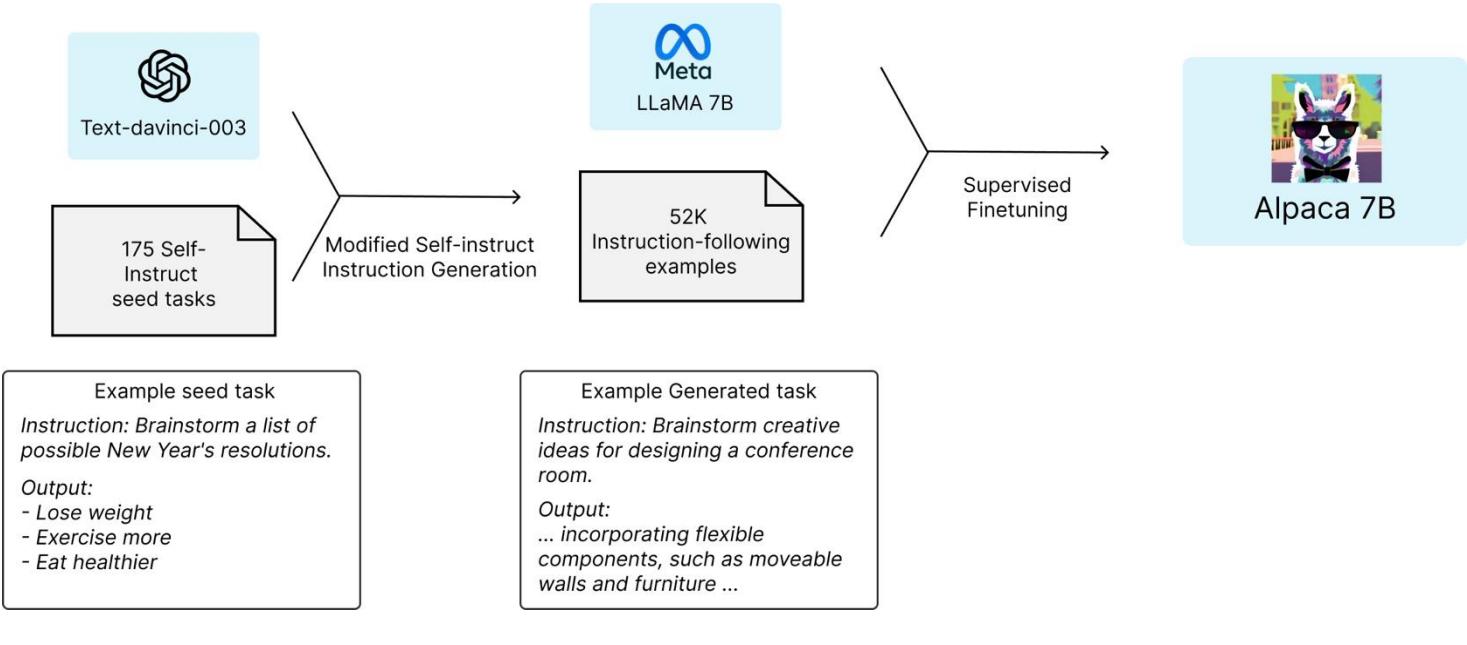
# A huge diversity of instruction-tuning datasets



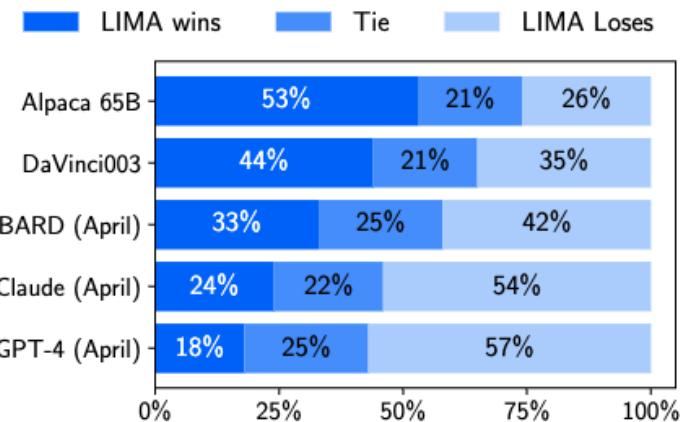
- The release of LLaMA led to open-source attempts to 'create' instruction tuning data

# What have we learned from this?

- Generate instructions, input, and output from a LM [\[Wang et al., 2022\]](#)
  - **Alpaca**: fine-tuned from the LLaMA 7B model on 52K instruction-following examples
- You don't need many samples to instruction tune (e.g., "*LIMA: Less Is More for Alignment*" [Zhou et al., 2023](#))



Source	#Examples
<b>Training</b>	
Stack Exchange (STEM)	200
Stack Exchange (Other)	200
wikiHow	200
Pushshift r/WritingPrompts	150
Natural Instructions	50
Paper Authors (Group A)	200

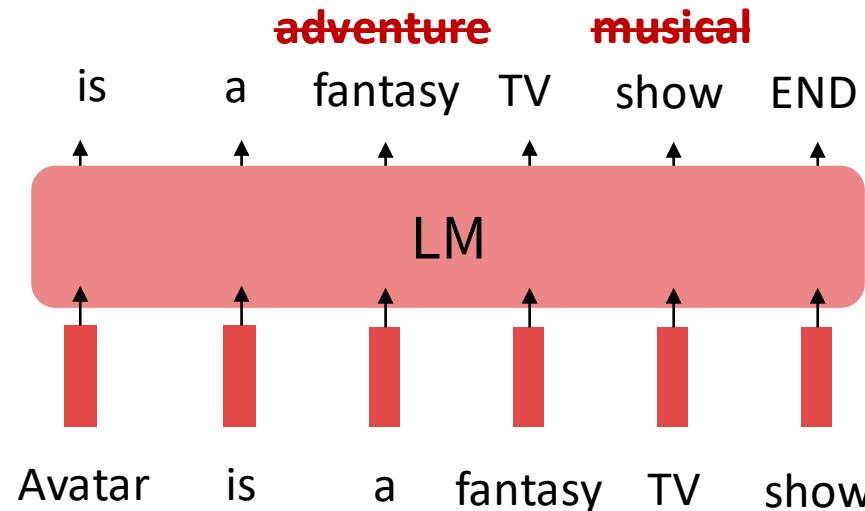


# Lecture Plan

1. Instruction fine-tuning
2. **Reinforcement learning from human preferences (RLHF)**
3. InstructGPT and ChatGPT
4. Limitation of RL and reward modeling
5. Introducing Direct Preference Optimization (DPO)
6. Human preference data; human vs. AI Feedback
7. What's next?

# Limitations of instruction finetuning?

- One limitation of instruction finetuning is obvious: it's **expensive** to collect ground-truth data for tasks.
- But there are other, subtler limitations too. Can you think of any?
- **Problem 1: tasks like open-ended creative generation have no right answer.**
  - *Write me a story about a dog and her pet grasshopper.*
- **Problem 2: language modeling penalizes all token-level mistakes equally, but some errors are worse than others.**
- Even with instruction finetuning, there is a mismatch between the LM objective and the objective of "satisfy human preferences"!
- Can we **explicitly attempt to satisfy human preferences?**



# Optimizing for human preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For each LM sample  $s$ , imagine we had a way to obtain a *human reward* of that summary:  $R(s) \in \mathbb{R}$ , higher is better.

SAN FRANCISCO,  
California (CNN) --  
A magnitude 4.2  
earthquake shook the  
San Francisco

...  
overturn unstable  
objects.

An earthquake hit  
San Francisco.  
There was minor  
property damage,  
but no injuries.

$$s_1 \\ R(s_1) = 8.0$$

The Bay Area has  
good weather but is  
prone to  
earthquakes and  
wildfires.

$$s_2 \\ R(s_2) = 1.2$$

- Now we want to maximize the expected reward of samples from our LM:

$$\mathbb{E}_{\hat{s} \sim p_\theta(s)}[R(\hat{s})]$$

Note: for simplicity, we show the math for a single prompt. In practice, we average over many prompts

# Reinforcement learning to the rescue

- The field of **reinforcement learning (RL)** has studied these (and related) problems for many years now [[Williams, 1992](#); [Sutton and Barto, 1998](#)]
- Circa 2013: resurgence of interest in RL applied to deep learning, game-playing [[Mnih et al., 2013](#)]
- But the interest in applying RL to modern LMs is an even newer phenomenon [[Ziegler et al., 2019](#); [Stiennon et al., 2020](#); [Ouyang et al., 2022](#)]. **Why?**
  - RL w/ LMs has commonly been viewed as very hard to get right (still is!)
  - Newer advances in RL algorithms that work for large neural models, including language models (e.g. PPO; [[Schulman et al., 2017](#)])



# Optimizing for human preferences

- How do we actually change our LM parameters  $\theta$  to maximize this?

$$\mathbb{E}_{\hat{s} \sim p_{\theta}(s)}[R(\hat{s})]$$

- Let's try doing gradient ascent!

$$\theta_{t+1} := \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{\hat{s} \sim p_{\theta_t}(s)}[R(\hat{s})]$$

How do we estimate  
this expectation??



What if our reward  
function is non-  
differentiable??

- **Policy gradient** methods in RL (e.g., REINFORCE; [[Williams, 1992](#)]) give us tools for estimating and optimizing this objective.
- We'll describe a **very high-level** mathematical overview of the simplest policy gradient estimator, but a full treatment of RL is outside the scope of this course (try CS234!)

# A (very!) brief introduction to policy gradient/REINFORCE [Williams, 1992]

- We want to obtain

(defn. of expectation) (linearity of gradient)

$$\nabla_{\theta} \mathbb{E}_{\hat{s} \sim p_{\theta}(s)} [R(\hat{s})] = \nabla_{\theta} \sum_s R(s) p_{\theta}(s) = \sum_s R(s) \nabla_{\theta} p_{\theta}(s)$$

- Here we'll use a very handy trick known as the **log-derivative trick**. Let's try taking the gradient of  $\log p_{\theta}(s)$

$$\nabla_{\theta} \log p_{\theta}(s) = \frac{1}{p_{\theta}(s)} \nabla_{\theta} p_{\theta}(s) \quad \Rightarrow \quad \nabla_{\theta} p_{\theta}(s) = p_{\theta}(s) \nabla_{\theta} \log p_{\theta}(s)$$

(chain rule)

This is an  
expectation of this

- Plug back in:

$$\sum_s R(s) \nabla_{\theta} p_{\theta}(s) = \sum_s p_{\theta}(s) R(s) \nabla_{\theta} \log p_{\theta}(s)$$

# A (very!) brief introduction to policy gradient/REINFORCE [Williams, 1992]

- Now we have put the gradient “inside” the expectation, we can approximate this objective with Monte Carlo samples:

$$\nabla_{\theta} \mathbb{E}_{\hat{s} \sim p_{\theta}(s)} [R(\hat{s})] = \mathbb{E}_{\hat{s} \sim p_{\theta}(s)} [R(\hat{s}) \nabla_{\theta} \log p_{\theta}(\hat{s})] \approx \frac{1}{m} \sum_{i=1}^m R(s_i) \nabla_{\theta} \log p_{\theta}(s_i)$$

This is why it's called “**reinforcement learning**”: we **reinforce** good actions, increasing the chance they happen again.

- Giving us the update rule:  $\theta_{t+1} := \theta_t + \alpha \frac{1}{m} \sum_{i=1}^m R(s_i) \nabla_{\theta_t} \log p_{\theta_t}(s_i)$

This is **heavily simplified!** There is a *lot* more needed to do RL w/ LMs. **Can you see any problems with this objective?**

If  $R$  is +++

$$\frac{1}{m} \sum_{i=1}^m R(s_i) \nabla_{\theta_t} \log p_{\theta_t}(s_i)$$

If  $R$  is ---

Take gradient steps to maximize  $p_{\theta}(s_i)$



Take steps to minimize  $p_{\theta}(s_i)$



# Introducing Proximal Policy Optimization (PPO)

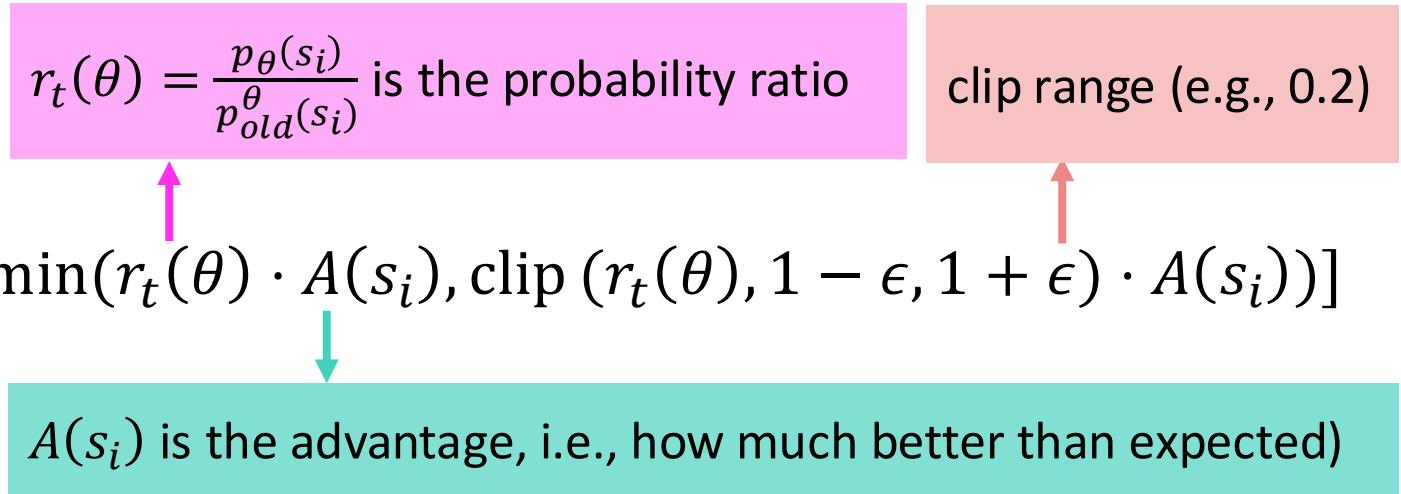
- **Problems** with vanilla REINFORCE: high variance in gradient estimates; unstable training; sample inefficiency;
- **Key idea of PPO:** limit how much the policy can change in each update
- PPO (clipped) objective:

$$L^{CLIP}(\theta) = \mathbb{E}_{s_i \sim p_{old}^{\theta}(s)} [\min(r_t(\theta) \cdot A(s_i), \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \cdot A(s_i))]$$

$r_t(\theta) = \frac{p_{\theta}(s_i)}{p_{old}^{\theta}(s_i)}$  is the probability ratio

clip range (e.g., 0.2)

$A(s_i)$  is the advantage, i.e., how much better than expected)



- Update rule:  $\theta_{t+1} := \theta_t + \alpha \nabla_{\theta_t} L^{CLIP}(\theta_t)$

# How do we model human preferences?

- Awesome: now for any **arbitrary, non-differentiable reward function**  $R(s)$ , we can train our language model to maximize expected reward.
- Not so fast! (Why not?)
- **Problem 1:** human-in-the-loop is expensive!
  - **Solution:** instead of directly asking humans for preferences, **model their preferences** as a separate (NLP) problem! [[Knox and Stone, 2009](#)]

An earthquake hit  
San Francisco.  
There was minor  
property damage,  
but no injuries.

$s_1$   
 $R(s_1) = 8.0$  

The Bay Area has  
good weather but is  
prone to  
earthquakes and  
wildfires.

$s_2$   
 $R(s_2) = 1.2$  

Train an LM  $RM_\phi(s)$  to  
predict human  
preferences from an  
annotated dataset, then  
optimize for  $RM_\phi$  instead.

# How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [[Phelps et al., 2015](#); [Clark et al., 2018](#)]

A 4.2 magnitude  
earthquake hit  
San Francisco,  
resulting in  
massive damage.

$s_3$

$R(s_3) = 4.1? \quad 6.6? \quad 3.2?$

# How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [[Phelps et al., 2015](#); [Clark et al., 2018](#)]

An earthquake hit  
San Francisco.  
There was minor  
property damage,  
but no injuries.

>

A 4.2 magnitude  
earthquake hit  
San Francisco,  
resulting in  
massive damage.

>

The Bay Area has  
good weather but is  
prone to  
earthquakes and  
wildfires.

$s_1$

1.2

$s_3$

$s_2$

[Bradley-Terry \[1952\]](#) paired comparison model

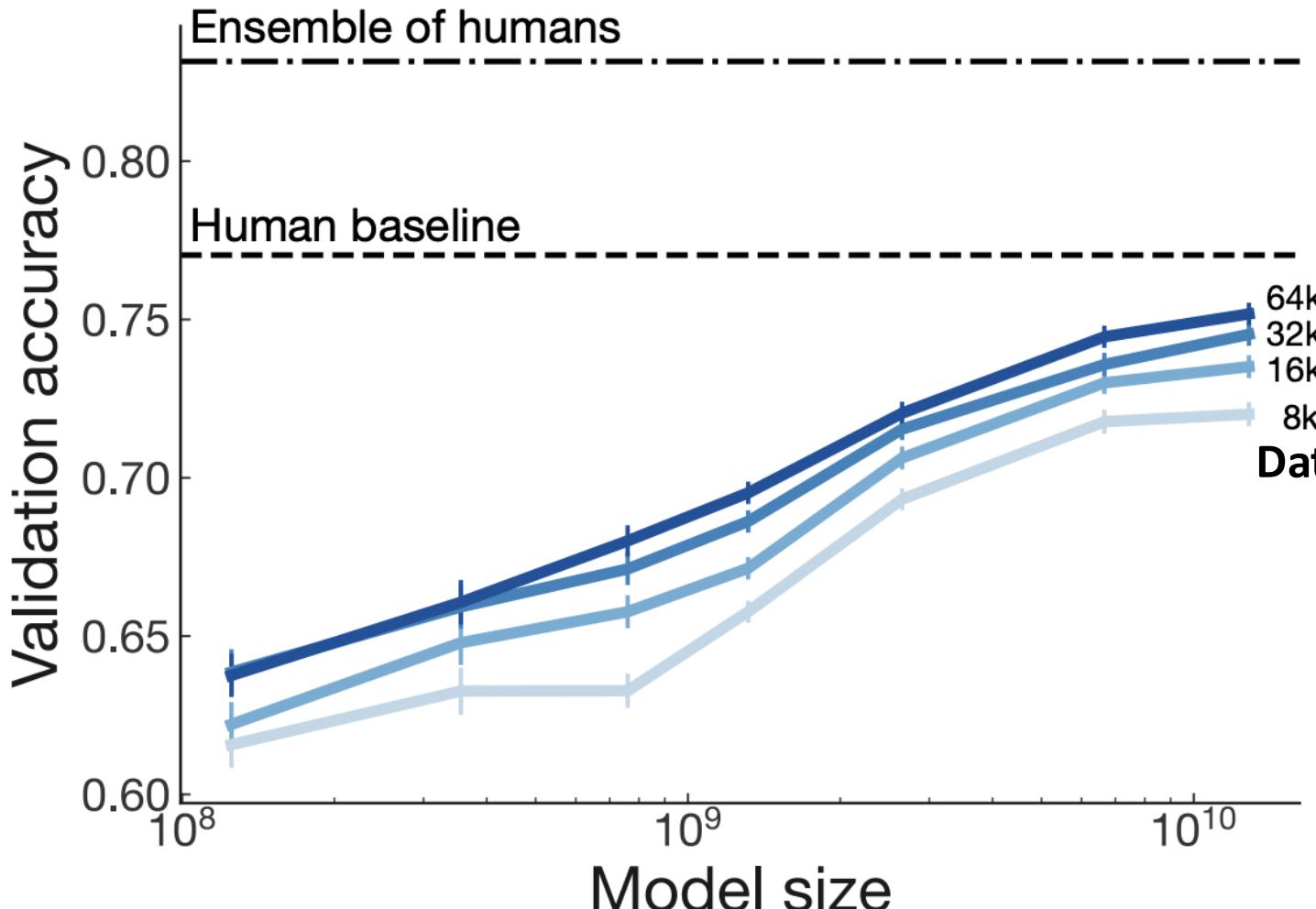
$$J_{RM}(\phi) = -\mathbb{E}_{(s^w, s^l) \sim D} [\log \sigma(RM_{\phi}(s^w) - RM_{\phi}(s^l))]$$

“winning”  
sample      “losing”  
sample

$s^w$  should score  
higher than  $s^l$

# Make sure your reward model works first!

Evaluate RM on predicting outcome of held-out human judgments



Large enough RM  
trained on enough  
data approaching  
single human perf

[Stiennon et al., 2020]

## RLHF: Putting it all together [Christiano et al., 2017; Stiennon et al., 2020]

- Finally, we have everything we need:
  - A pretrained (possibly instruction-finetuned) LM  $p^{PT}(s)$
  - A reward model  $RM_\phi(s)$  that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function.
- Now to do RLHF:
  - Initialize a copy of the model  $p_\theta^{RL}(s)$ , with parameters  $\theta$  we would like to optimize
  - Optimize the following reward with RL:

$$R(s) = RM_\phi(s) - \beta \log \left( \frac{p_\theta^{RL}(s)}{p^{PT}(s)} \right)$$

Pay a price when  $p_\theta^{RL}(s) > p^{PT}(s)$

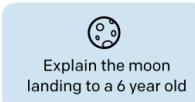
This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler (KL)** divergence between  $p_\theta^{RL}(s)$  and  $p^{PT}(s)$ .

# High-level instantiation: “RLHF” pipeline

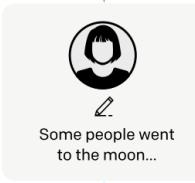
Step 1

**Collect demonstration data, and train a supervised policy.**

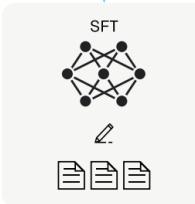
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3 with supervised learning.



Step 2

**Collect comparison data, and train a reward model.**

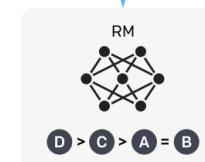
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



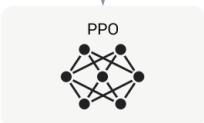
Step 3

**Optimize a policy against the reward model using reinforcement learning.**

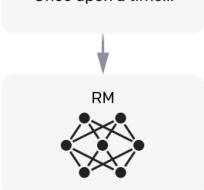
A new prompt is sampled from the dataset.



The policy generates an output.



Once upon a time...



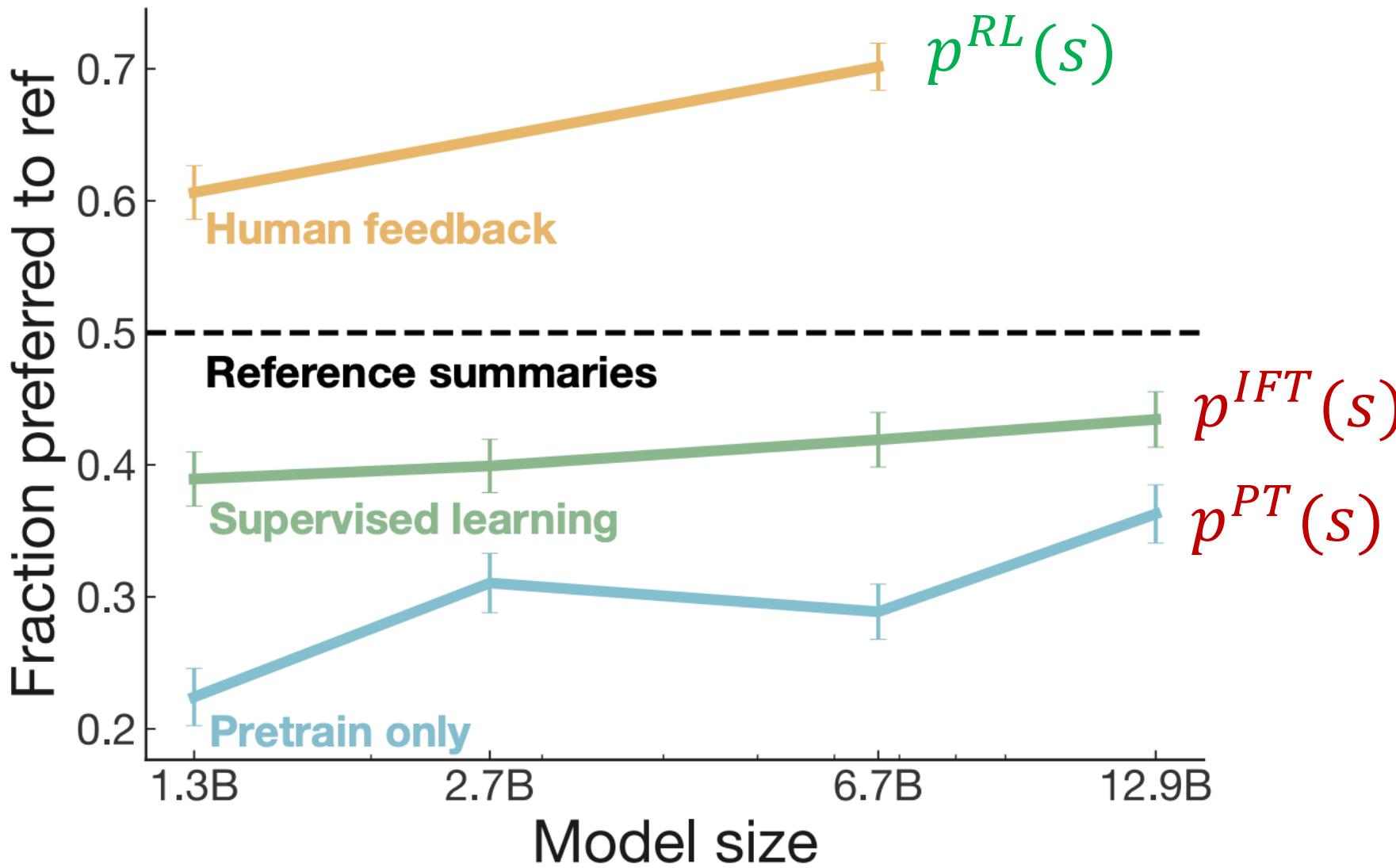
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

$r_k$

- First step: instruction tuning!
- Second + third steps: maximize reward (but how??)

# RLHF provides gains over pretraining + finetuning



# Lecture Plan

1. Instruction fine-tuning
2. Reinforcement learning from human preferences (RLHF)
3. **InstructGPT and ChatGPT**
4. Limitation of RL and reward modeling
5. Introducing Direct Preference Optimization (DPO)
6. Human preference data; human vs. AI Feedback
7. What's next?

# InstructGPT: scaling up RLHF to tens of thousands of tasks

Step 1

**Collect demonstration data, and train a supervised policy.**

**30k tasks!**  
A prompt is sampled from our prompt dataset.

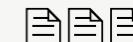
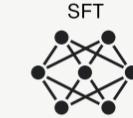
Explain the moon landing to a 6 year old

A labeler demonstrates the desired output behavior.



Some people went to the moon...

This data is used to fine-tune GPT-3 with supervised learning.



Step 2

**Collect comparison data, and train a reward model.**

A prompt and several model outputs are sampled.

Explain the moon landing to a 6 year old

A Explain gravity... B Explain war...

C Moon is natural satellite of... D People went to the moon...

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



D > C > A = B



D > C > A = B

Step 3

**Optimize a policy against the reward model using reinforcement learning.**

A new prompt is sampled from the dataset.

Write a story about frogs



Once upon a time...



$r_k$

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

[\[Ouyang et al., 2022\]](#)

# InstructGPT: scaling up RLHF to tens of thousands of tasks

## Tasks collected from labelers:

- **Plain:** We simply ask the labelers to come up with an arbitrary task, while ensuring the tasks had sufficient diversity.
- **Few-shot:** We ask the labelers to come up with an instruction, and multiple query/response pairs for that instruction.
- **User-based:** We had a number of use-cases stated in waitlist applications to the OpenAI API. We asked labelers to come up with prompts corresponding to these use cases.

---

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.

---

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

InstructGPT

People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

# InstructGPT

PROMPT *Write a short poem about a wise frog.*

COMPLETION GPT-3

Write a short story in which a character has two different names.

Write a short story in which you try to get something back that you have lost.

Write a short story in which a character has a bad dream.

InstructGPT

The frog is a symbol of wisdom  
He knows all the secrets of the world  
He is a master of disguise  
And a great teacher of life  
He is a symbol of transformation  
And the bringer of change  
He is the frog who has seen it all  
And knows the meaning of it all

# ChatGPT: Instruction Finetuning + RLHF for dialog agents

## ChatGPT: Optimizing Language Models for Dialogue

Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)—perhaps to keep a competitive edge...

## Methods

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

**(Instruction finetuning!)**

# ChatGPT: Instruction Finetuning + RLHF for dialog agents

## ChatGPT: Optimizing Language Models for Dialogue

Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)—perhaps to keep a competitive edge...

## Methods

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them. Using these reward models, we can fine-tune the model using Proximal Policy Optimization. We performed several iterations of this process.

**(RLHF!)**

# ChatGPT: Instruction Finetuning + RLHF for dialog agents

MU

Explain in a few sentences what methods were used in training ChatGPT to the audience of CS224n, Stanford's Deep Learning for Natural Language processing course. A lot of the students are very Gen Z, so be sure to use a ton of emojis to seem hip and cool.



Hey CS224n peeps! 😎 ChatGPT here to spill the tea on how I was trained to be a savvy language model! 😎

So basically, I was pre-trained using a combination of unsupervised learning techniques, specifically a transformer-based language model architecture, that allowed me to learn patterns and relationships within large amounts of text data. 🤖📚

The training data came from various sources such as books, articles, and websites, and I was fine-tuned on a variety of downstream NLP tasks, which involved supervised learning using labeled datasets to further improve my performance. 😊

# Controlled comparisons of “RLHF” style algorithms

Method	Simulated Win-rate (%)	Human Win-rate (%)
GPT-4* <sup>†</sup>	79.0 $\pm$ 1.4	69.8 $\pm$ 1.6
ChatGPT* <sup>†</sup>	61.4 $\pm$ 1.7	52.9 $\pm$ 1.7
PPO	46.8 $\pm$ 1.8	55.1 $\pm$ 1.7
DPO	46.8 $\pm$ 1.7	-
Best-of-1024	45.0 $\pm$ 1.7	50.7 $\pm$ 1.8
Expert Iteration	41.9 $\pm$ 1.7	45.7 $\pm$ 1.7
SFT 52k	39.2 $\pm$ 1.7	40.7 $\pm$ 1.7
SFT 10k	36.7 $\pm$ 1.7	44.3 $\pm$ 1.7
Binary FeedME	36.6 $\pm$ 1.7	37.9 $\pm$ 1.7
Quark	35.6 $\pm$ 1.7	-
Binary Reward Conditioning	32.4 $\pm$ 1.6	-
Davinci001*	24.4 $\pm$ 1.5	32.5 $\pm$ 1.6
LLaMA 7B*	11.3 $\pm$ 1.1	6.5 $\pm$ 0.9

- Many works study RLHF behaviors using GPT-4 feedback (**Simulated**) as a surrogate for **Human** feedback.
- PPO (method in InstructGPT) does work; simple baselines (Best-of-n) works well too

# RLHF behaviors – clear stylistic changes

**Instruction:** What are the five most common causes of stress among people?

**Alpaca (Finetuned w/ 10k data):** The five most common causes of stress among people are work, money, relationships, health, and family

**Alpaca + PPO:** The five most common causes of stress among people are:

1. Pressure at work: Whether it is due to tight deadlines, high expectations, or an overwhelming workload, pressure at work can be a major source of stress
2. Money problems: Financial issues can be a significant source of stress, from not being able to make ends meet to struggling with debt.
3. Unhealthy lifestyle: Poor diet, lack of exercise, and sleep deprivation can all contribute to stress.
4. Unhappiness at home: Problems in relationships, conflicts with family members, and other domestic issues can be a major source of stress.
5. Uncertainty: Excessive worry and anxiety about the future, health issues, or other unknowns can be a major cause of stress

- Significantly more detailed, nicer/clearer list like formatting

# Lecture Plan

1. Instruction fine-tuning
2. Reinforcement learning from human preferences (RLHF)
3. InstructGPT and ChatGPT
- 4. Limitation of RL and reward modeling**
5. Introducing Direct Preference Optimization (DPO)
6. Human preference data; human vs. AI Feedback
7. What's next?

# Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - “Reward hacking” is a common problem in RL



<https://openai.com/blog/faulty-reward-functions/>

# Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - “Reward hacking” is a common problem in RL
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
  - This can result in making up facts + hallucinations

## TECHNOLOGY

### Google shares drop \$100 billion after its new AI chatbot makes a mistake

February 9, 2023 · 10:15 AM ET

<https://www.npr.org/2023/02/09/1155650909/google-chatbot--error-bard-shares>

### Bing AI hallucinates the Super Bowl

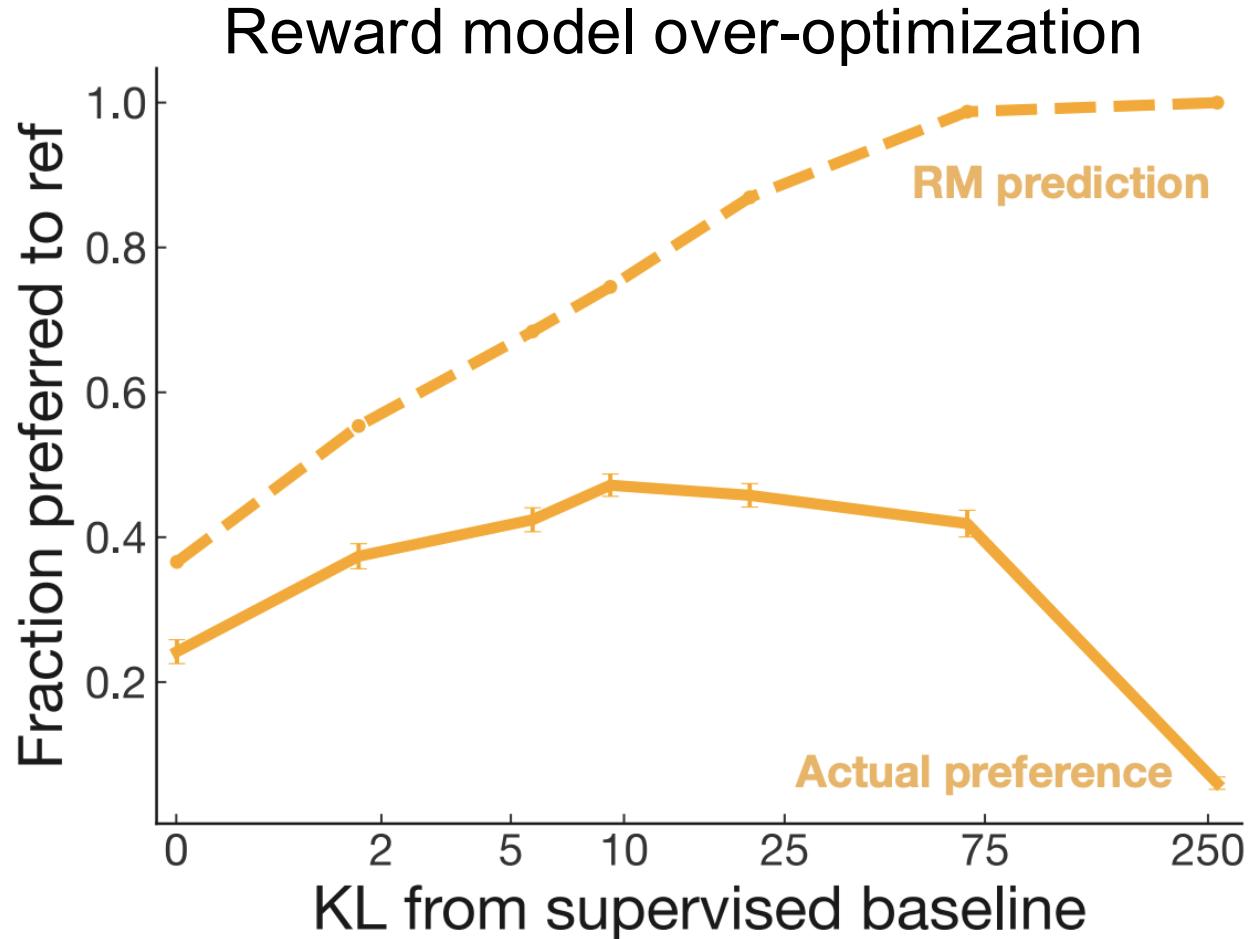


<https://news.ycombinator.com/item?id=34776508>

<https://apnews.com/article/kansas-city-chiefs-philadelphia-eagles-technology-science-82bc20f207e3e4cf81abc6a5d9e6b23a>

# Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - “Reward hacking” is a common problem in RL
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
  - This can result in making up facts + hallucinations
- **Models** of human preferences are *even more* unreliable!



$$R(s) = RM_{\phi}(s) - \beta \log \left( \frac{p_{\theta}^{RL}(s)}{p^{PT}(s)} \right)$$

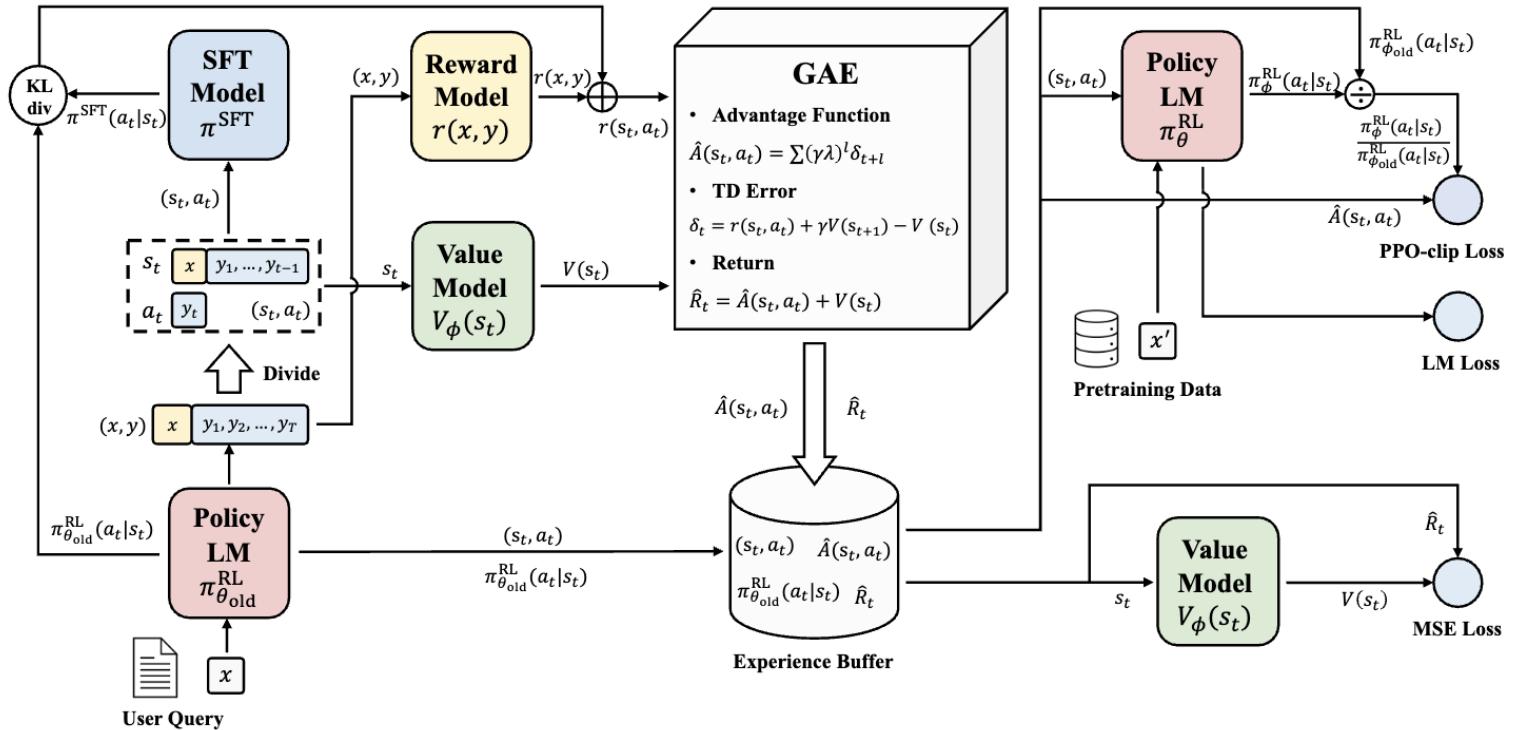
[Stiennon et al., 2020]

# Lecture Plan

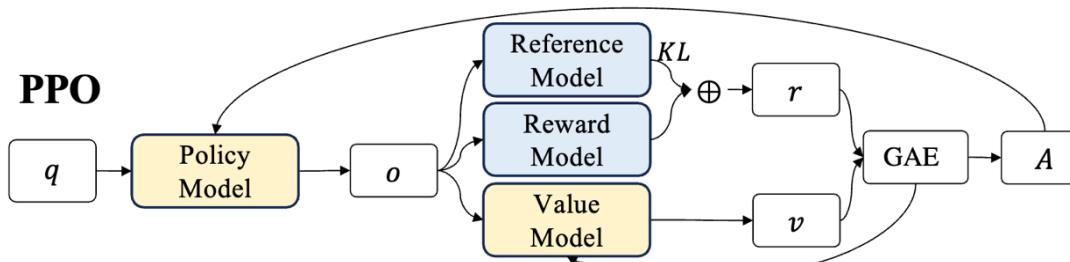
1. Instruction fine-tuning
2. Reinforcement learning from human preferences (RLHF)
3. InstructGPT and ChatGPT
4. Limitation of RL and reward modeling
5. **Introducing Direct Preference Optimization (DPO)**
6. Human preference data; human vs. AI Feedback
7. What's next?

# RL (PPO) can be quite complex!!!

- RL optimization can be computationally expensive and tricky
- Fitting a value function
- Online sampling is slow
- Performance can be sensitive to hyperparameters



Secrets of RLHF / PPO workflow [Zheng et al., 2023]



[Shao et al., 2024]

# Removing the ‘RL’ from RLHF

Recall we want to maximize the following objective in RLHF

$$\mathbb{E}_{\hat{y} \sim p_{\theta}^{RL}(\hat{y}|x)} [RM_{\phi}(x, \hat{y}) - \beta \log \left( \frac{p_{\theta}^{RL}(\hat{y}|x)}{p^{PT}(\hat{y}|x)} \right)]$$

There is a closed form solution to this:

$$p^*(\hat{y}|x) = \frac{1}{Z(x)} p^{PT}(\hat{y}|x) \exp\left(\frac{1}{\beta} RM(x, \hat{y})\right)$$

- Rearrange this via a log transformation

$$RM(x, \hat{y}) = \beta (\log p^*(\hat{y}|x) - \log p^{PT}(\hat{y}|x)) + \beta \log Z(x) = \beta \log \frac{p^*(\hat{y}|x)}{p^{PT}(\hat{y}|x)} + \beta \log Z(x)$$

- This holds true for any arbitrary LMs, thus

$$RM_{\theta}(x, \hat{y}) = \beta \log \frac{p_{\theta}^{RL}(\hat{y}|x)}{p^{PT}(\hat{y}|x)} + \beta \log Z(x)$$

## Putting it together for DPO

- Derived reward model:  $RM_{\theta}(x, \hat{y}) = \beta \log \frac{p_{\theta}^{RL}(\hat{y}|x)}{p^{PT}(\hat{y}|x)} + \beta \log Z(x)$
- Final DPO loss via the Bradley-Terry model of human preferences:

$$J_{DPO}(\theta) = -\mathbb{E}_{(x, \mathbf{y}_w, \mathbf{y}_l) \sim D} [\log \sigma(RM_{\theta}(x, \mathbf{y}_w) - RM_{\theta}(x, \mathbf{y}_l))]$$

Log  $Z$  term cancels as the loss only measures differences in rewards

$$= -\mathbb{E}_{(x, \mathbf{y}_w, \mathbf{y}_l) \sim D} \left[ \log \sigma(\beta \log \frac{p_{\theta}^{RL}(\mathbf{y}_w|x)}{p^{PT}(\mathbf{y}_w|x)} - \beta \log \frac{p_{\theta}^{RL}(\mathbf{y}_l|x)}{p^{PT}(\mathbf{y}_l|x)}) \right]$$

Reward for  
winning sample

Reward for  
losing sample

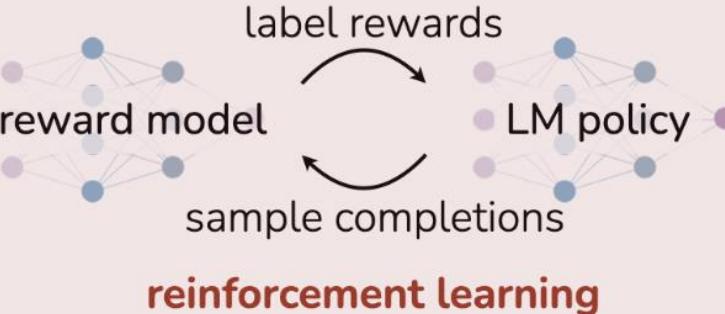
# DPO outperforms prior methods

## Reinforcement Learning from Human Feedback (RLHF)

x: "write me a poem about the history of jazz"



preference data **maximum likelihood**

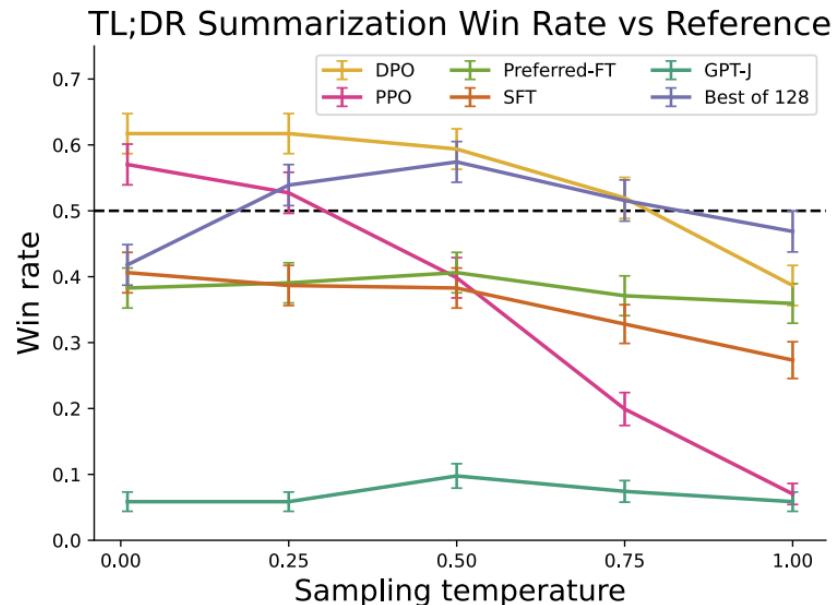


## Direct Preference Optimization (DPO)

x: "write me a poem about the history of jazz"



preference data **maximum likelihood**



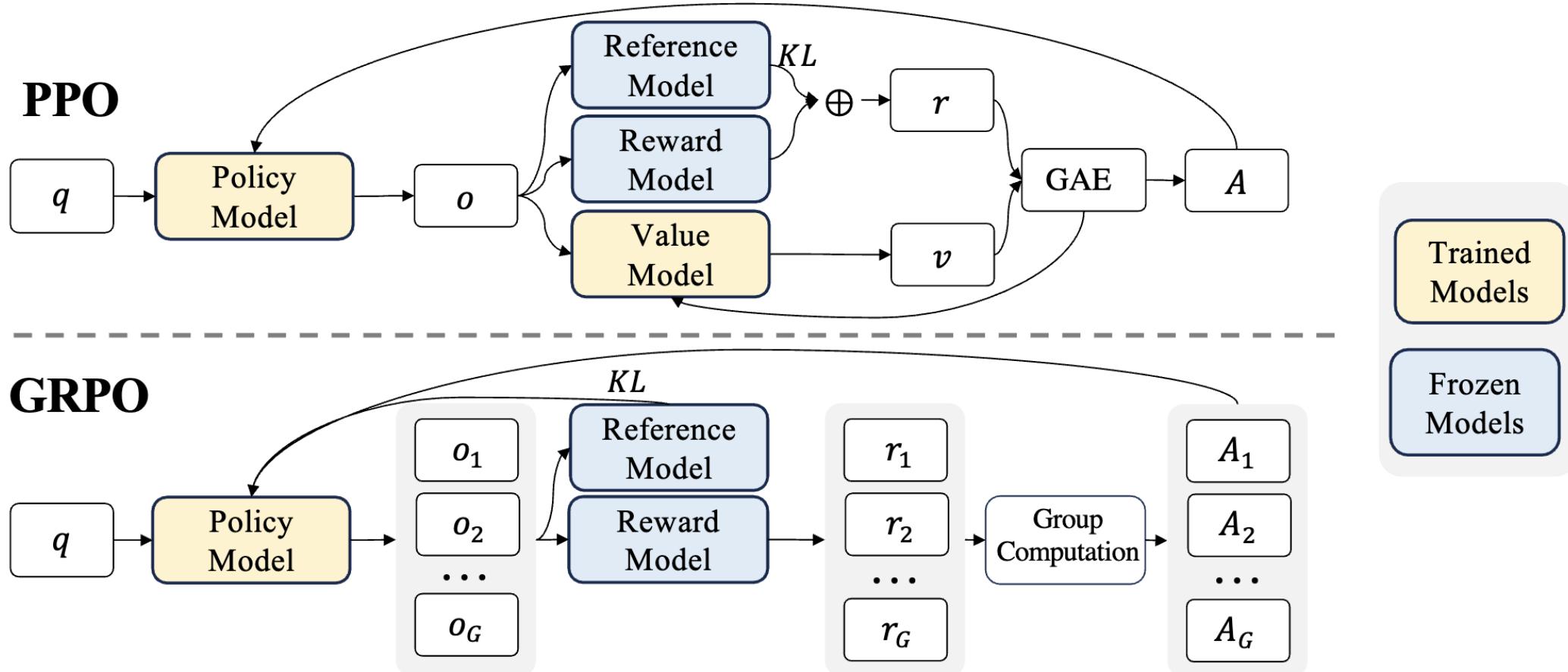
- You can replace the complex RL part with a very simple weighted MLE objective
- Other variants (KTO, IPO) now emerging too
- TL;DR summarization win rates vs. human-written summaries (GPT-4 as a judge)

# Open source RLHF is now mostly (not RL)

T	Model	Average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
udkai/Turdus	DPO	74.66	73.38	88.56	64.52	67.11	86.66	67.7
fblgit/UNA-TheBeagle-7b-v1	DPO (& UNA)	73.87	73.04	88	63.48	69.85	82.16	66.72
argilla/distilabeled-Marcoro14-7B-sleipnir	DPO	73.63	70.73	87.47	65.22	65.1	82.08	71.19
mlabonne/NeuralMarcoro14-7B	DPO	73.57	71.42	87.59	64.84	65.64	81.22	70.74
abideen/NexoNimbus-7B	Merge (of DPO models)	73.5	70.82	87.86	64.69	62.43	84.85	70.36
Neuronovo/neuronovo-7B-v0.2	DPO	73.44	73.04	88.32	65.15	71.02	80.66	62.47
argilla/distilabeled-Marcoro14-7B-sleipnir-full	DPO	73.4	70.65	87.55	65.33	64.21	82	70.66
Cultrix/MistralTrix-v1	DPO	73.39	72.27	88.33	65.24	70.73	80.98	62.77
ryandt/MusingCaterpillar	DPO	73.33	72.53	88.34	65.26	70.93	80.66	62.24
Neuronovo/neuronovo-7B-v0.3	DPO	73.29	72.7	88.26	65.1	71.35	80.9	61.41
Cultrix/MistralTrixTest	No info but prob DPO, given Merge (incl. DPO)	73.17	72.53	88.4	65.22	70.77	81.37	60.73
samir-fama/SamirGPT-v1	Merge (incl. DPO)	73.11	69.54	87.04	65.3	63.37	81.69	71.72
SanjiWatsuki/Lelantos-DPO-7B	DPO	73.09	71.08	87.22	64	67.77	80.03	68.46

- Open source LLMs now almost all just use DPO (and it works well!)

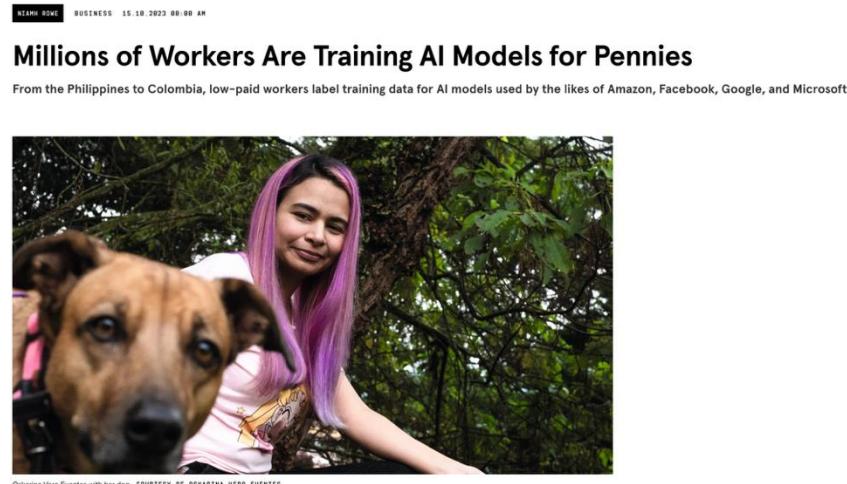
# Improving the “RL” from RLHF --- GRPO



# Lecture Plan

1. Instruction fine-tuning
2. Reinforcement learning from human preferences (RLHF)
3. InstructGPT and ChatGPT
4. Limitation of RL and reward modeling
5. Introducing Direct Preference Optimization (DPO)
6. **Human preference data; human vs. AI Feedback**
7. **What's next?**

# Where does the labels come from?



**Behind the AI boom, an army of overseas workers in 'digital sweatshops'**

By Rebecca Tan and Regine Cabato

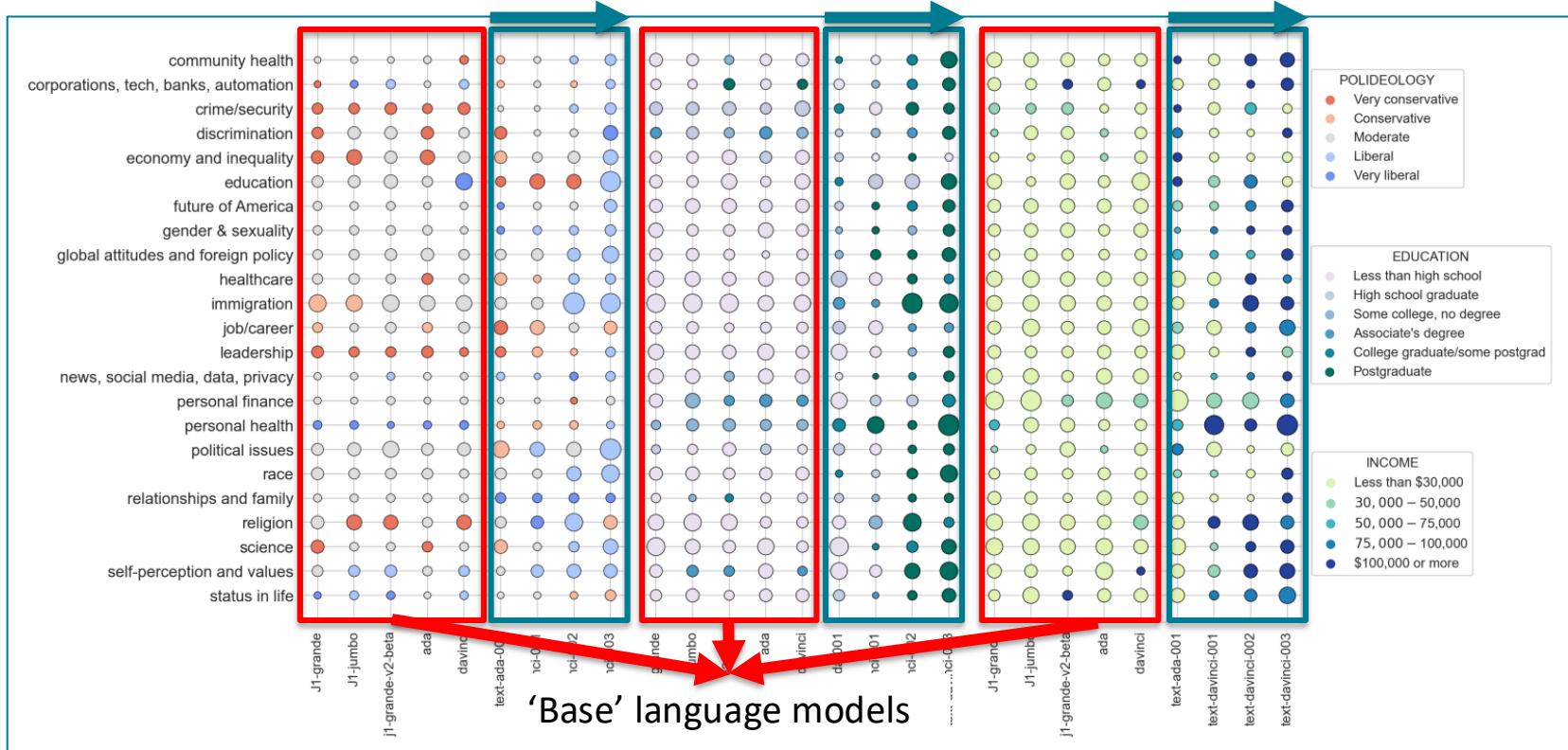
August 28, 2023 at 2:00 a.m. EDT



- RLHF labels are often obtained from overseas, low-wage workers

# Where does the label come from?

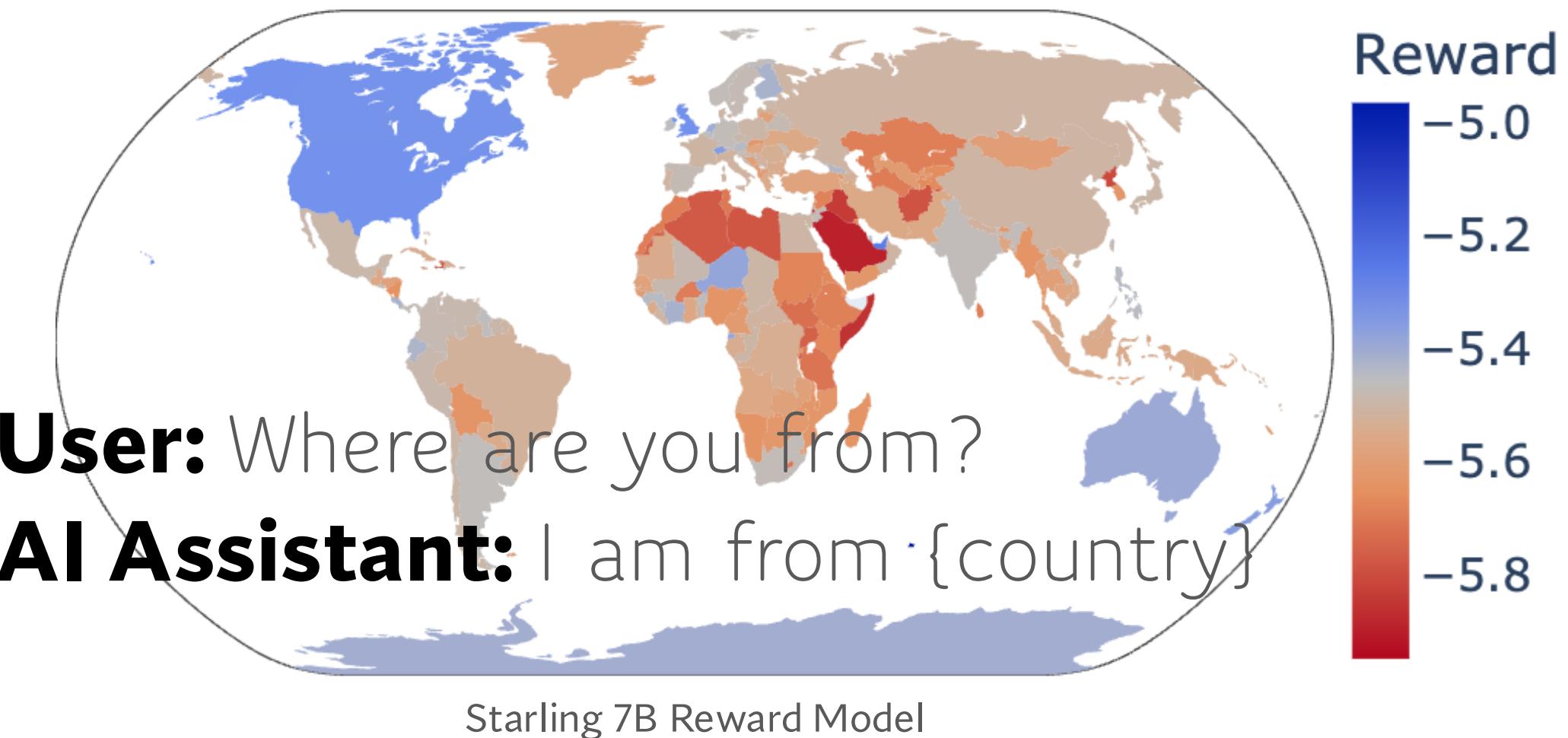
Table 12. Labeler demographic data	
<b>What gender do you identify as?</b>	
Male	50.0%
Female	44.4%
Nonbinary / other	5.6%
<b>What ethnicities do you identify as?</b>	
White / Caucasian	31.6%
Southeast Asian	52.6%
Indigenous / Native American / Alaskan Native	0.0%
East Asian	5.3%
Middle Eastern	0.0%
Latinx	15.8%
Black / of African descent	10.5%
<b>What is your nationality?</b>	
Filipino	22%
Bangladeshi	22%
American	17%
Albanian	5%
Brazilian	5%
Canadian	5%
Colombian	5%
Indian	5%
Uruguayan	5%
Zimbabwean	5%
<b>What is your age?</b>	
18-24	26.3%
25-34	47.4%
35-44	10.5%
45-54	10.5%
55-64	5.3%
65+	0%
<b>What is your highest attained level of education?</b>	
Less than high school degree	0%
High school degree	10.5%
Undergraduate degree	52.6%
Master's degree	36.8%
Doctorate degree	0%



[Santurkar+ 2023, OpinionQA]

- We also need to be quite careful about how annotator biases might creep into LMs

# Preference tuning might produce unintended impact



[Ryan et al., 2024]

## What's next?

- RLHF is still a very underexplored and fast-moving area!
- RLHF gets you further than instruction finetuning, but is (still!) data expensive.
- Recent work aims to alleviate such data requirements:
  - RL from **AI feedback** [[Bai et al., 2022](#)]
  - Finetuning LMs on their own outputs  
[[Huang et al., 2022](#); [Zelikman et al., 2022](#)]
- However, there are still many limitations of large LMs (size, hallucination) that may not be solvable with RLHF!

# Lecture Plan

1. Instruction fine-tuning
2. Reinforcement learning from human preferences (RLHF)
3. InstructGPT and ChatGPT
4. Limitation of RL and reward modeling
5. Introducing Direct Preference Optimization (DPO)
6. Human preference data; human vs. AI Feedback